



## PRESTO : Mode d'emploi

Denis Talay

### ► To cite this version:

Denis Talay. PRESTO : Mode d'emploi. [Rapport de recherche] RT-0106, INRIA. 1989, pp.43. inria-00070060

**HAL Id: inria-00070060**

**<https://inria.hal.science/inria-00070060>**

Submitted on 19 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNITÉ DE RECHERCHE  
INRIA-SOPHIA ANTIPOLIS

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
B.P. 105  
78153 Le Chesnay Cedex  
France  
Tél. (1) 39 63 55 11

# Rapports Techniques

N° 106

*Programme 5*

**PRESTO :  
MODE D'EMPLOI**

**Denis TALAY**

**Mars 1989**



# PRESTO : MODE D'EMPLOI

Denis TALAY

INRIA  
Avenue Emile Hugues  
Sophia Antipolis  
F-06565 Valbonne (France)

## Résumé

Ce document est le manuel d'utilisation de **Presto**, un générateur automatique de programmes Fortran résolvant numériquement les Systèmes Différentiels Stochastiques.

## Abstract

This document is the **Presto** reference manual. **Presto** is a generator of Fortran programs numerically solving the Stochastic Differential Systems.

## Table des matières

<b>1</b>	<b>But de Presto</b>	<b>4</b>
<b>2</b>	<b>Organisation interne</b>	<b>5</b>
2.1	Présentation . . . . .	5
2.2	L'interface-utilisateurs . . . . .	5
2.3	La connaissance . . . . .	5
2.4	Les procédures Reduce . . . . .	7
<b>3</b>	<b>Installation et appel</b>	<b>8</b>
<b>4</b>	<b>Exemple de programme engendré</b>	<b>8</b>
<b>5</b>	<b>Aspects mathématiques</b>	<b>16</b>
<b>6</b>	<b>Mode d'emploi</b>	<b>17</b>
6.1	Le Menu Principal . . . . .	19
6.2	Rentrée des données . . . . .	20
6.3	Caractéristiques du système : dialogue interactif . . . . .	21
6.3.1	Dimension du système . . . . .	21
6.3.2	Dimension du bruit . . . . .	23
6.3.3	Rentrée des coefficients . . . . .	23
6.4	Caractéristiques du système : fichier Reduce . . . . .	25
6.5	Ito ou Stratonovich . . . . .	26
6.6	But des simulations . . . . .	26
6.7	Temps d'intégration numérique . . . . .	27
6.8	Loi initiale . . . . .	27
6.9	Générateur de nombres aléatoires . . . . .	29
6.10	Fonction à intégrer . . . . .	30
6.11	Nombre de simulations . . . . .	31
6.12	Schéma de discrétisation . . . . .	32
6.13	Pas de discrétisation . . . . .	32

6.14 Génération du programme Fortran . . . . .	33
6.15 Consultation et modification . . . . .	34
6.16 Paramètres . . . . .	35
6.17 Fichiers Fortran . . . . .	36
6.18 Sauvegarde des données . . . . .	38
6.19 Passage par Reduce et Lisp . . . . .	39
6.20 Procédure de relancement . . . . .	40

## 1 But de Presto

**Presto** a pour but d'engendrer automatiquement des programmes Fortran résolvant numériquement des systèmes dynamiques excités par des bruits aléatoires multiplicatifs éventuellement multidimensionnels, de la forme :

$$\frac{d}{dt}X(t) = b(X(t)) + s(X(t))\xi(t)$$

Dans sa première version, **Presto** limite son savoir-faire au cas où  $(\xi(t))$  est un bruit blanc. Le système précédent s'écrit alors naturellement sous la forme d'un système différentiel stochastique au sens de Stratonovich. Pour certaines applications, en Mécanique notamment, ces modélisations au sens de Stratonovich sont très naturelles, a contrario le calcul stochastique (et d'autres applications) préfèrent les systèmes écrits au sens d'Ito (nous verrons plus loin que **Presto** effectue de lui-même la transformation qui permet de réécrire au sens d'Ito un système au sens de Stratonovich).

Par conséquent, **Presto** traite les systèmes différentiels suivants, où  $(W(t))$  est un processus de Wiener, qu'ils soient compris au sens d'Ito ou de Stratonovich (ce sera précisé lors de la session).

$$dX(t) = b(X(t))dt + s(X(t))dW(t) \quad (1)$$

L'utilisateur de **Presto** n'est pas supposé connaître la théorie des processus stochastiques, ni leur analyse numérique, ni les techniques de simulation du hasard (et des intégrales stochastiques particulièrement), ni même à la limite Fortran. **Presto** lui demande seulement une description minimale de son problème : le dialogue s'effectue de manière conviviale à l'aide d'une souris, un clavier et un écran bitmap.

La description du problème achevée, **Presto**

- analyse les données afin de choisir, par exemple, un bon algorithme de discrétisation parmi ceux qu'il "connait";
- effectue tous les calculs algébriques préalables à l'écriture du programme (en utilisant les fonctionnalités du système de calcul formel Reduce) ;
- engendre un programme Fortran commenté ;
- engendre un fichier de données contenant les valeurs des diverses variables Fortran que l'on peut vouloir modifier ad libitum sans avoir à recompiler le programme lui-même : paramètres imposés par l'utilisateur concernant les coefficients de (1), pas de discrétisation, temps d'intégration, etc.

Ecrit entièrement en *Le\_Lisp*, et incluant la traduction de Reduce en *Le\_Lisp* (due à Alain BEGES), **Presto** est aisément portable et accepte aussi bien X-windows

que Suntools. Il est actuellement en service au C.W.I. (Pays Bas), à l'I.D.H.E.A.P. (Université de Lausanne), à l'Université de Clermont, à l'Ecole des Ponts, etc.

La conception d'ensemble et une ébauche rustique du système ont été réalisées avec l'aide efficace de **Juliette Leblond** (et décrites dans Leblond & Talay [15]).

Pour l'interface-utilisateurs, j'ai bénéficié du concours d'Anne-Marie Bustos, Bruno Chabrier et Frédérique Kerlen.

## 2 Organisation interne

### 2.1 Présentation

**Presto** n'est pas le premier logiciel combinant expertise d'un problème exposé de manière mathématique, calcul formel et génération automatique de programmes. Ainsi, Chancelier, Gomez, Quadrat & Sulem [7] ont développé un système-expert pour la commande, l'optimisation et l'identification de systèmes dynamiques stochastiques : **Pandore**.

Comme tous ces logiciels, **Presto** est constitué d'une interface-utilisateurs, une représentation de sa connaissance, un algorithme de consultation de cette connaissance, et un ensemble de procédures écrites dans le langage de calcul formel choisi.

### 2.2 L'interface-utilisateurs

Sans originalité, l'interface-utilisateurs se présente sous la forme de menus (où cliquer avec une souris) et de fenêtres de saisie. Un filtre d'événements permet à tout instant de ne déclencher une action que si s'est produit un événement précis (par exemple, clic-souris dans une région déterminée de l'écran).

On saisit par clavier des objets de deux natures différentes :

- les quantités de type impérativement numérique (dimension du processus, valeur du pas de discrétisation,...) ou chaîne de caractères (nom du fichier de sauvegarde des données,...) : elles sont lues à l'aide de l'interprète de *Le.Lisp* ;
- les quantités de type impérativement algébrique (expression analytique des coefficients du système,...) : elles sont saisies à l'aide du lecteur de *Reduce* ; ceci impose des conventions particulières, détaillées plus loin.

### 2.3 La connaissance

Un soin particulier a été apporté à la représentation de la connaissance. En effet, l'analyse numérique des systèmes différentiels stochastiques en est à ses débuts et, par ailleurs, il semblait raisonnable de limiter nos ambitions dans un premier temps : ainsi

ont été exclus (pour l'instant) les modèles à bruits à large bande, quelques méthodes de discrétisation, les raffinements des méthodes de Monte-Carlo, etc. Nous voulions donc pouvoir faire évoluer **Presto** progressivement.

Une autre contrainte était d'utiliser un seul langage (*Le\_Lisp*). Ceci excluait de recourir à Prolog, par exemple.

Enfin, les programmes que **Presto** doit engendrer peuvent être considérés comme des assemblages de "chapitres" distincts, le contenu Fortran de chaque chapitre dépendant de paramètres variables : par exemple, "dimension du système (1) et dépendance en temps des coefficients", ou bien "générateur de nombres pseudo-aléatoires choisi par l'utilisateur" ; le nombre des contenus possibles varient entre 1 et 10, par contre, un même contenu apparaîtra dans plusieurs situations distinctes. (Signalons qu'au contraire, **Pandore** résout le problème de commande par diverses méthodes, chacune ayant son domaine d'application privilégié et une mise en oeuvre particulière : la programmation dynamique et la méthode du gradient stochastique se programment très différemment).

Plus précisément, l'architecture du programme Fortran à engendrer dépend seulement du but de la résolution numérique du système (1) (simulation de trajectoires de  $X(t)$ , calcul de moments de  $X(t)$ , etc : nous précisons les buts traités actuellement plus loin).

A chacun de ces buts sont associés deux objets :

- le "contexte", liste Lisp de tous les renseignements dont **Presto** pourra avoir besoin pour analyser la description du problème de l'utilisateur : concrètement, ses éléments sont des symboles que **Presto** rassemble lors du dialogue interactif (le système est-il défini au sens de Ito ou de Stratonovich, etc), ou par appel d'une procédure **Reduce** si des calculs algébriques sont nécessaires (test de commutativité portant sur les vecteurs-colonnes de la matrice  $s$  décrit plus loin, etc), ou enfin par déduction ; ainsi, la présence de "Sito" signifie que le système (1) a été défini au sens d'Ito, celle de "Scommut" signifie que le test de commutativité a été satisfait, etc ;
- le "plan du programme Fortran", liste Lisp de symboles (qui désignent les "chapitres" du programme) ; ses éléments seront, par exemple :
  - comment (pour : commentaires rappelant les expressions algébriques des coefficients du système (1) et le but de la résolution numérique)
  - data (pour : lecture de données numériques dans un fichier de données)
  - constantes (pour : calcul de diverses constantes "universelles" ( $\pi$ , etc))
  - init-sto (pour : initialisation des constantes liées à la simulation des intégrales stochastiques)
  - x0 (pour : simulation de la condition initiale  $X(0)$ )
  - debut-boucles (pour : début des boucles des itérations)



- integrales-stochastiques (pour : simulation des intégrales stochastiques apparaissant dans le schéma de discrétisation)
- schema (pour : mise en oeuvre du schéma de discrétisation)
- fin-boucles (pour : fin des boucles des itérations)
- fin (pour : fin du programme)

Chaque symbole est le nom d'une nouvelle liste, qui représente un arbre de décision : chaque feuille est le nom d'une procédure Reduce dont l'exécution produit un des contenus possibles du chapitre correspondant à l'arbre (et effectue les calculs algébriques nécessaires à cette génération) ; chaque noeud d'une branche est un élément possible de "contexte", et représente donc une condition nécessaire pour que soit appelée la procédure Reduce indiquée par la feuille de la branche.

Après la phase de dialogue, **Presto** établit en premier le "plan du programme Fortran", puis parcourt les arbres membres du "plan", en cherchant pour chacun si une branche est composée de noeuds tous membres de "contexte" ; la feuille d'une telle branche fournit le nom d'une procédure Reduce, l'exécution séquentielle de toutes les procédures ainsi sélectionnées produit le programme Fortran complet.

La modification de la connaissance est simple : l'apprentissage d'une nouvelle méthode de discrétisation se traduit par une modification de l'arbre "schéma" (afin que soient décrites les conditions d'application de la méthode en question), et par l'écriture de la procédure Reduce engendrant le code Fortran correspondant à la nouvelle méthode.

## 2.4 Les procédures Reduce

Les procédures Reduce de **Presto** ont été écrites dans le langage algébrique ou symbolique de Reduce, puis toutes traduites en *Le\_Lisp* grâce à l'option "defn" de Reduce.

Parmi les procédures symboliques, signalons une redéfinition du printer d'expressions Fortran : lorsque l'on veut engendrer un programme écrit en double précision, il est important que toutes les constantes numériques (autres que les indices, les étiquettes, les dimensions de tableaux) soient écrites sous la forme 1.0d+00. De manière standard, Reduce offre au mieux la possibilité de remplacer 1 par 1., ce qui n'est pas suffisant. Notre nouveau printer effectue de lui-même la conversion en double précision dans toutes les expressions algébriques qui lui sont soumises (sous réserve, bien sûr, que l'utilisateur ait choisi l'option "double précision" dans le menu "type des constantes du programme Fortran"). Entre autres précautions, il a fallu modifier le compteur de caractères d'impression, (indispensable puisque les lignes Fortran sont limitées à 72 caractères, et qu'en conséquence une expression trop longue doit être écrite à l'aide de cartes-suite.

### 3 Installation et appel

**Presto** est installé sous la forme de trois fichiers (outre LELISPBIN et LELISPGO qui permettent seulement d'appeler *Le\_Lisp*) : le script-shell PRESTO (respectivement PRESTO-X) crée une fenêtre Graphics-Tool dans l'environnement SUNTOOLS (respectivement xterm dans l'environnement X) et y exécute le script-shell PRESTO-SUB (respectivement PRESTO-SUB-X), ce dernier appelant *Le\_Lisp* en restaurant le fichier PRESTO.core, qui contient à la fois Reduce (version *Le\_Lisp*), et le code de **Presto**, y compris la traduction de ses procédures algébriques et symboliques en *Le\_Lisp*.

Avant le premier lancement, il suffit de modifier le fichier PRESTO (respectivement PRESTO-X) de telle sorte que la variable shell DIRPRESTO soit initialisée au nom de la directory d'accueil des fichiers de PRESTO.

L'appel de **Presto** s'effectue alors simplement par le chargement de l'environnement bitmap de son choix, puis par la commande "PRESTO" (respectivement "PRESTO-X").

### 4 Exemple de programme engendré

Voici un exemple de programme engendré par **Presto**.

Les commentaires en tête du programme énoncent le problème qu'il s'agit de résoudre.

```
c the aim is to compute the expectation of f(x(t)),
c where the function f is defined by :
c
c x1**2.0
c
c and (x(t)) is the diffusion solution of
c
c dx(t) = b(t,x(t))dt + s(t,x(t))dw(t)
c
c with :
c
c the vector-valued function b defined by :
c b(1)=(3.0*sqrt(2.0)*x1-2.0*x1*sin(omega*t)*omega-12.0*x1+6.0*
c sqrt(2.0)*x2-6.0*x2)/(4.0*(cos(omega*t)+2.0))
c b(2)=(6.0*sqrt(2.0)*x1-6.0*x1+3.0*sqrt(2.0)*x2-2.0*x2*sin(
c omega*t)*omega-12.0*x2)/(4.0*(cos(omega*t)+2.0))
c
```

```

c s(1,1)=sin(vnu*(x1+x2))
c s(1,2)=cos(vnu*(x1+x2))
c s(2,1)=sin((3.0*x1*vnu+3.0*x2*vnu+pi)/3.0)
c s(2,2)=cos((3.0*x1*vnu+3.0*x2*vnu+pi)/3.0)
c
c   initializations
c   -----
c       implicit double precision(a-h,o-z)
c       character*50 texte
c
c       dimension tabx(2,1000)
c
c opening of the file which will contain the results
c       close(11)
c       open(11,form='formatted',file='prg.res')
c
c reading of the parameters of the program :
c discretization step, integration time,
c number of simulated independent paths,
c initialization of the random generator
c       close(12)
c       open(12,form='formatted',file='prg.data')
c       rewind(12)
c       read(12,*)texte
c       read(12,*)h
c       read(12,*)texte
c       read(12,*)texte
c       read(12,*)tmax
c       read(12,*)texte
c       read(12,*)texte
c       read(12,*)nsimu
c       read(12,*)texte
c       read(12,*)texte
c       read(12,*)nseed
c       read(12,*)texte
c       read(12,*)texte
c       read(12,*)omega
c       read(12,*)texte
c       read(12,*)texte
c       read(12,*)vnu
c
c       write(11,*)'temps d integration   = ',tmax
c       write(11,*)'pas de discretisation = ',h
c       write(11,*)'nombre de simulations = ',nsimu

```

```

        write(11,*)'valeur de omega = ',omega
        write(11,*)'valeur de vnu = ',vnu
c
c constants useful for the simulation
    pi=atan(1.0d+00)*4.0d+00
    xdpi=2.0d+00*pi
    xrh=sqrt(h)
    t=0.0d+00
c
c initialization of the random generator
    call g05cbf(nseed)
c
c x(0)
    do 10 nj=1,nsimu
c
        tr1=g05caf(tr1)
        tr2=g05caf(tr2)
        tar1=sqrt(-2.0d+00*log(tr1))
        tar2=xdpi*tr2
        xgaus1=tar1*sin(tar2)
        xgaus2=tar1*cos(tar2)
c
        x1=xgaus1
        x2=(xgaus1+xgaus2)/sqrt(2.0d+00)
c
        tabx(1,nj)=x1
        tabx(2,nj)=x2
c
10    continue
c
c number of iterations
    niter=tmax/h+1
c
c simulations
c -----
    do 20 nk=1,niter
c
        xsum=0.0d+00
c
        do 200 nj=1,nsimu
c
            x1=tabx(1,nj)
            x2=tabx(2,nj)
c

```

```

c simulation of the stochastic integrals
  tr1=g05caf(tr1)
  tr2=g05caf(tr2)
  tar1=sqrt(-2.0d+00*log(tr1))
  tar2=xdpi*tr2
  dw1=tar1*sin(tar2)
  dw2=tar1*cos(tar2)
c
  dw1=xrh*dw1
  dw2=xrh*dw2
c
  xs11=(dw1*dw1-h)/2.0d+00
  xs22=(dw2*dw2-h)/2.0d+00
c
  tr1=g05caf(tr1)
  xu12=-h
  if(tr1.le..5d+00) xu12=h
c
  xs12=(dw1*dw2+xu12)/2.0d+00
  xs21=(dw1*dw2-xu12)/2.0d+00
c
c the scheme used below is the montecarlo scheme ;
c for its description, see d.talay :
c "discretisation d'une e.d.s. et calcul approche
c d'esperances de fonctionnelles de la solution",
c m2an 20-1,141,1986,
c or e.pardoux-d.talay :
c "discretization and simulation of s.d.e.",
c acta applicandae mathematicae 3,23,1985.
c
c computation of x((p+1)h)
c
c discretization corresponding to the time dependency
  tt=t
c
  xy1=x1
  xy2=x2
c
  xrk11=(3.0d+00*sqrt(2.0d+00)*xy1+6.0d+00*sqrt(2.0d+00)*
. xy2-2.0d+00*sin(omega*t)*omega*xy1-12.0d+00*xy1-
. 6.0d+00*xy2)/(4.0d+00*(cos(omega*t)+2.0d+00))
  xrk12=(6.0d+00*sqrt(2.0d+00)*xy1+3.0d+00*sqrt(2.0d+00)*
. xy2-2.0d+00*sin(omega*t)*omega*xy2-6.0d+00*xy1-
. 12.0d+00*xy2)/(4.0d+00*(cos(omega*t)+2.0d+00))

```

c

```

t=tt+h/2.0d+00
xy1=x1+xrk11*h/2.0d+00
xy2=x2+xrk12*h/2.0d+00
xrk21=(3.0d+00*sqrt(2.0d+00)*xy1+6.0d+00*sqrt(2.0d+00)*
. xy2-2.0d+00*sin(omega*t)*omega*xy1-12.0d+00*xy1-
. 6.0d+00*xy2)/(4.0d+00*(cos(omega*t)+2.0d+00))
xrk22=(6.0d+00*sqrt(2.0d+00)*xy1+3.0d+00*sqrt(2.0d+00)*
. xy2-2.0d+00*sin(omega*t)*omega*xy2-6.0d+00*xy1-
. 12.0d+00*xy2)/(4.0d+00*(cos(omega*t)+2.0d+00))

```

c

```

xy1=x1+xrk21*h/2.0d+00
xy2=x2+xrk22*h/2.0d+00
xrk31=(3.0d+00*sqrt(2.0d+00)*xy1+6.0d+00*sqrt(2.0d+00)*
. xy2-2.0d+00*sin(omega*t)*omega*xy1-12.0d+00*xy1-
. 6.0d+00*xy2)/(4.0d+00*(cos(omega*t)+2.0d+00))
xrk32=(6.0d+00*sqrt(2.0d+00)*xy1+3.0d+00*sqrt(2.0d+00)*
. xy2-2.0d+00*sin(omega*t)*omega*xy2-6.0d+00*xy1-
. 12.0d+00*xy2)/(4.0d+00*(cos(omega*t)+2.0d+00))

```

c

```

t=tt+h
xy1=x1+xrk31*h
xy2=x2+xrk32*h
xrk41=(3.0d+00*sqrt(2.0d+00)*xy1+6.0d+00*sqrt(2.0d+00)*
. xy2-2.0d+00*sin(omega*t)*omega*xy1-12.0d+00*xy1-
. 6.0d+00*xy2)/(4.0d+00*(cos(omega*t)+2.0d+00))
xrk42=(6.0d+00*sqrt(2.0d+00)*xy1+3.0d+00*sqrt(2.0d+00)*
. xy2-2.0d+00*sin(omega*t)*omega*xy2-6.0d+00*xy1-
. 12.0d+00*xy2)/(4.0d+00*(cos(omega*t)+2.0d+00))

```

c

```

xy1=(xrk11+2.0d+00*xrk21+2.0d+00*xrk31+xrk41)*h/6.0d+00
xy2=(xrk12+2.0d+00*xrk22+2.0d+00*xrk32+xrk42)*h/6.0d+00
t=tt

```

c

```

xq1=2.0d+00**((1.0d+00/2.0d+00)
xq2=cos(vnu*(x1+x2))
xq3=cos((3.0d+00*x1*vnu+3.0d+00*x2*vnu+pi)/3.0d+00)
xq4=sin(vnu*(x1+x2))
xq5=sin((3.0d+00*x1*vnu+3.0d+00*x2*vnu+pi)/3.0d+00)
xq6=cos(omega*t)
xq7=sin(omega*t)
ans06=-16.0d+00*xs22*vnu*xq3*xq4
ans05=8.0d+00*xs11*vnu*xq2*xq4*xq6+16.0d+00*xs11*vnu*
. xq2*xq4+8.0d+00*xs11*vnu*xq2*xq5*xq6+16.0d+00*xs11*

```

```

. vnu*xq2*xq5-8.0d+00*xs12*vnu*xq4**2.0d+00*xq6-
. 16.0d+00*xs12*vnu*xq4**2.0d+00-8.0d+00*xs12*vnu*xq4*xq5*xq6
. -16.0d+00*xs12*vnu*xq4*xq5+8.0d+00*xs21*vnu*xq2**
. 2.0d+00*xq6+16.0d+00*xs21*vnu*xq2**2.0d+00+8.0d+00*xs21*vnu
. *xq2*xq3*xq6+16.0d+00*xs21*vnu*xq2*xq3-8.0d+00*xs22*
. vnu*xq2*xq4*xq6-16.0d+00*xs22*vnu*xq2*xq4-8.0d+00*
. xs22*vnu*xq3*xq4*xq6+ans06
ans04=-4.0d+00*dw2*h*vnu**2.0d+00*xq2*xq3**2.0d+00-
. 2.0d+00*dw2*h*vnu**2.0d+00*xq2*xq4**2.0d+00*xq6-4.0d+00*dw2
. *h*vnu**2.0d+00*xq2*xq4**2.0d+00-4.0d+00*dw2*h*vnu**
. 2.0d+00*xq2*xq4*xq5*xq6-8.0d+00*dw2*h*vnu**2.0d+00*xq2*xq4*
. xq5-2.0d+00*dw2*h*vnu**2.0d+00*xq2*xq5**2.0d+00*xq6-
. 4.0d+00*dw2*h*vnu**2.0d+00*xq2*xq5**2.0d+00+3.0d+00*dw2*h*
. xq1*xq2+6.0d+00*dw2*h*xq1*xq3-12.0d+00*dw2*h*xq2-
. 6.0d+00*dw2*h*xq3+8.0d+00*dw2*xq2*xq6+16.0d+00*dw2*xq2+
. ans05
ans03=-4.0d+00*dw1*h*vnu**2.0d+00*xq4*xq5**2.0d+00+
. 3.0d+00*dw1*h*xq1*xq4+6.0d+00*dw1*h*xq1*xq5-12.0d+00*dw1*h*
. xq4-6.0d+00*dw1*h*xq5+8.0d+00*dw1*xq4*xq6+16.0d+00*
. dw1*xq4-2.0d+00*dw2*h*omega*xq2*xq7-2.0d+00*dw2*h*vnu
. **2.0d+00*xq2**3.0d+00*xq6-4.0d+00*dw2*h*vnu**2.0d+00
. *xq2**3.0d+00-4.0d+00*dw2*h*vnu**2.0d+00*xq2**2.0d+00
. *xq3*xq6-8.0d+00*dw2*h*vnu**2.0d+00*xq2**2.0d+00*xq3-
. 2.0d+00*dw2*h*vnu**2.0d+00*xq2*xq3**2.0d+00*xq6+ans04
ans02=-4.0d+00*dw1*h*vnu**2.0d+00*xq2**2.0d+00*xq4-
. 4.0d+00*dw1*h*vnu**2.0d+00*xq2*xq3*xq4*xq6-8.0d+00*dw1*h*
. vnu**2.0d+00*xq2*xq3*xq4-2.0d+00*dw1*h*vnu**2.0d+00*
. xq3**2.0d+00*xq4*xq6-4.0d+00*dw1*h*vnu**2.0d+00*xq3**
. 2.0d+00*xq4-2.0d+00*dw1*h*vnu**2.0d+00*xq4**3.0d+00*xq6-
. 4.0d+00*dw1*h*vnu**2.0d+00*xq4**3.0d+00-4.0d+00*dw1*h*vnu**
. 2.0d+00*xq4**2.0d+00*xq5*xq6-8.0d+00*dw1*h*vnu**2.0d+00*xq4
. **2.0d+00*xq5-2.0d+00*dw1*h*vnu**2.0d+00*xq4*xq5**
. 2.0d+00*xq6+ans03
ans01=-2.0d+00*x1*dw1*h*omega*vnu*xq2*xq7+9.0d+00*x1*
. dw1*h*vnu*xq1*xq2-18.0d+00*x1*dw1*h*vnu*xq2+2.0d+00*
. x1*dw2*h*omega*vnu*xq4*xq7-9.0d+00*x1*dw2*h*vnu*xq1*
. xq4+18.0d+00*x1*dw2*h*vnu*xq4-2.0d+00*x2*dw1*h*omega*
. vnu*xq2*xq7+9.0d+00*x2*dw1*h*vnu*xq1*xq2-18.0d+00*x2*
. dw1*h*vnu*xq2+2.0d+00*x2*dw2*h*omega*vnu*xq4*xq7-
. 9.0d+00*x2*dw2*h*vnu*xq1*xq4+18.0d+00*x2*dw2*h*vnu*xq4-
. 2.0d+00*dw1*h*omega*xq4*xq7-2.0d+00*dw1*h*vnu**2.0d+00*xq2
. **2.0d+00*xq4*xq6+ans02
xx1=ans01/(8.0d+00*(xq6+2.0d+00))
ans06=-16.0d+00*xs22*vnu*xq3*xq5

```

```

ans05=8.0d+00*xs11*vnu*xq3*xq4*xq6+16.0d+00*xs11*vnu*
. xq3*xq4+8.0d+00*xs11*vnu*xq3*xq5*xq6+16.0d+00*xs11*
. vnu*xq3*xq5-8.0d+00*xs12*vnu*xq4*xq5*xq6-16.0d+00*
. xs12*vnu*xq4*xq5-8.0d+00*xs12*vnu*xq5**2.0d+00*xq6-
. 16.0d+00*xs12*vnu*xq5**2.0d+00+8.0d+00*xs21*vnu*xq2*xq3*xq6
. +16.0d+00*xs21*vnu*xq2*xq3+8.0d+00*xs21*vnu*xq3**
. 2.0d+00*xq6+16.0d+00*xs21*vnu*xq3**2.0d+00-8.0d+00*xs22*vnu
. *xq2*xq5*xq6-16.0d+00*xs22*vnu*xq2*xq5-8.0d+00*xs22*
. vnu*xq3*xq5*xq6+ans06
ans04=-4.0d+00*dw2*h*vnu**2.0d+00*xq3**3.0d+00-2.0d+00*
. dw2*h*vnu**2.0d+00*xq3*xq4**2.0d+00*xq6-4.0d+00*dw2*h
. *vnu**2.0d+00*xq3*xq4**2.0d+00-4.0d+00*dw2*h*vnu**
. 2.0d+00*xq3*xq4*xq5*xq6-8.0d+00*dw2*h*vnu**2.0d+00*xq3*xq4*
. xq5-2.0d+00*dw2*h*vnu**2.0d+00*xq3*xq5**2.0d+00*xq6-
. 4.0d+00*dw2*h*vnu**2.0d+00*xq3*xq5**2.0d+00+6.0d+00*dw2*h*
. xq1*xq2+3.0d+00*dw2*h*xq1*xq3-6.0d+00*dw2*h*xq2-
. 12.0d+00*dw2*h*xq3+8.0d+00*dw2*xq3*xq6+16.0d+00*dw2*xq3+
. ans05
ans03=-4.0d+00*dw1*h*vnu**2.0d+00*xq5**3.0d+00+6.0d+00*
. dw1*h*xq1*xq4+3.0d+00*dw1*h*xq1*xq5-6.0d+00*dw1*h*xq4
. -12.0d+00*dw1*h*xq5+8.0d+00*dw1*xq5*xq6+16.0d+00*dw1*
. xq5-2.0d+00*dw2*h*omega*xq3*xq7-2.0d+00*dw2*h*vnu**
. 2.0d+00*xq2**2.0d+00*xq3*xq6-4.0d+00*dw2*h*vnu**2.0d+00*xq2
. **2.0d+00*xq3-4.0d+00*dw2*h*vnu**2.0d+00*xq2*xq3**
. 2.0d+00*xq6-8.0d+00*dw2*h*vnu**2.0d+00*xq2*xq3**2.0d+00-
. 2.0d+00*dw2*h*vnu**2.0d+00*xq3**3.0d+00*xq6+ans04
ans02=-4.0d+00*dw1*h*vnu**2.0d+00*xq2**2.0d+00*xq5-
. 4.0d+00*dw1*h*vnu**2.0d+00*xq2*xq3*xq5*xq6-8.0d+00*dw1*h*
. vnu**2.0d+00*xq2*xq3*xq5-2.0d+00*dw1*h*vnu**2.0d+00*
. xq3**2.0d+00*xq5*xq6-4.0d+00*dw1*h*vnu**2.0d+00*xq3**
. 2.0d+00*xq5-2.0d+00*dw1*h*vnu**2.0d+00*xq4**2.0d+00*xq5*xq6
. -4.0d+00*dw1*h*vnu**2.0d+00*xq4**2.0d+00*xq5-4.0d+00*
. dw1*h*vnu**2.0d+00*xq4*xq5**2.0d+00*xq6-8.0d+00*dw1*h
. *vnu**2.0d+00*xq4*xq5**2.0d+00-2.0d+00*dw1*h*vnu**
. 2.0d+00*xq5**3.0d+00*xq6+ans03
ans01=-2.0d+00*x1*dw1*h*omega*vnu*xq3*xq7+9.0d+00*x1*
. dw1*h*vnu*xq1*xq3-18.0d+00*x1*dw1*h*vnu*xq3+2.0d+00*
. x1*dw2*h*omega*vnu*xq5*xq7-9.0d+00*x1*dw2*h*vnu*xq1*
. xq5+18.0d+00*x1*dw2*h*vnu*xq5-2.0d+00*x2*dw1*h*omega*
. vnu*xq3*xq7+9.0d+00*x2*dw1*h*vnu*xq1*xq3-18.0d+00*x2*
. dw1*h*vnu*xq3+2.0d+00*x2*dw2*h*omega*vnu*xq5*xq7-
. 9.0d+00*x2*dw2*h*vnu*xq1*xq5+18.0d+00*x2*dw2*h*vnu*xq5-
. 2.0d+00*dw1*h*omega*xq5*xq7-2.0d+00*dw1*h*vnu**2.0d+00*xq2
. **2.0d+00*xq5*xq6+ans02

```



```

xx2=ans01/(8.0d+00*(xq6+2.0d+00))
c
x1=x1+xx1
x2=x2+xx2
c
x1=x1+xy1
x2=x2+xy2
c
xsum=xsum+(x1**2.0d+00)/nsimu
c
tabx(1,nj)=x1
tabx(2,nj)=x2
c
200 continue
c
t=t+h
write(11,*)'t = ',t,' ef(x(t)) = ',xsum
c
20 continue
c
c end of the program
c -----
c end

```

En même temps que ce programme, Presto a engendré le fichier de données Fortran suivant ("prg.data") :

```

c discretization step
0.1
c
c integration time
1
c
c number of simulated independent paths
1000
c
c initialization of the random generator
666666
c
c omega
1
c
c vnu
1

```

c

A l'exécution, le programme lira dans le fichier de données "prg.data" le pas de discrétisation, le nombre de simulations, etc, et créera le fichier "prg.res" dans lequel aura été écrite la valeur calculée de  $E|X^1(t)|^2$  pour chaque instant de la discrétisation.

Des modifications personnalisées de Presto sont possibles (si par exemple, l'utilisateur souhaite obtenir les résultats sous la forme d'une courbe GKS).

## 5 Aspects mathématiques

La plupart des méthodes de discrétisation utilisées par Presto actuellement sont décrites et analysées numériquement dans un article de compilation sur l'analyse numérique des systèmes différentiels stochastiques (Talay [34]), où on peut trouver également les énoncés théoriques qui donnent leurs vitesses de convergence respectives selon divers critères de convergence, et permettent donc de définir leurs domaines de validité respectifs. La bibliographie donnée par la référence ci-dessus a été recopiée à la fin de ce manuel.

Il apparaît notamment que la qualité théorique d'un schéma de discrétisation de (1) dépend du but de la résolution numérique : tel schéma qui diverge trajectoire par trajectoire est de vitesse de convergence élevée pour l'approximation de la loi de  $(X(t))$ .

Les buts pour lesquels la version actuelle de Presto peut apporter une aide, (et pour lesquels une étude de divers schémas a été conduite), sont les suivants :

1. simulation de la trajectoire de  $(X(t))$  correspondant à une trajectoire donnée de  $(W(t))$
2. calcul de  $Ef(X(t))$  sur un intervalle de temps fini
3. calcul de la loi invariante de  $(X(t))$  dans le cas où  $(X(t))$  est ergodique
4. calcul du plus grand exposant de Lyapunov dans le cas où le système (1) est bilinéaire

Soulignons que, dans le cas de bruits multidimensionnels, le plus souvent aucun schéma de discrétisation ne dépendant que des trajectoires de  $(W(t))$  ne converge trajectoire par trajectoire, et qu'un schéma convergent (le schéma de Milshtein) incluerait des intégrales stochastiques qui semblent impossibles à simuler exactement sur ordinateur, et dont il semble peu réaliste de supposer qu'on peut disposer de leurs trajectoires en pratique.

Néanmoins, il est possible de remplacer ces intégrales stochastiques par d'autres variables aléatoires convenablement choisies, de lois gaussiennes ou discrètes, de telle sorte que le nouveau schéma ainsi obtenu à partir du schéma de Milshtein soit d'ordre

sorte que le nouveau schéma ainsi obtenu à partir du schéma de Milshtein soit d'ordre 1 (i.e l'erreur qu'il induit peut être bornée par  $C^{te}h$ ,  $h$  étant le pas de discrétisation) pour les problèmes (2), (3), (4).

Des raffinements de ce schéma permettent ensuite de construire des schémas d'ordre au moins 2.

En outre, le système (1) doit être écrit au sens d'Ito pour être discrétisé efficacement (du moins si l'on veut obtenir un schéma explicite).

Lorsque l'utilisateur l'a défini au sens de Stratonovich, Presto effectue la transformation Stratonovich/Ito, ce qui se traduit par le calcul du nouveau coefficient de dérive

$$b_0(\cdot) = b(\cdot) + \frac{1}{2} \sum_{j=1}^r \partial s_j(\cdot) s_j(\cdot)$$

où  $s_j$  désigne la colonne  $j$  de la matrice  $s$ , et  $\partial s_j$  est la matrice dont l'élément de la ligne  $i$  et de la colonne  $k$  est  $\partial_k s_j^i$ .

Enfin, le plus souvent, les calculs numériques seront simplifiés si est vérifiée la condition de commutativité suivante (qui a aussi un rôle théorique crucial lorsque l'on veut obtenir des trajectoires de  $(X(t))$ ) :

$$\forall j, \forall k : \partial s_j(\cdot) s_k(\cdot) = \partial s_k(\cdot) s_j(\cdot)$$

Ce test est donc effectué, et les simplifications dues à son succès sont prises en compte lors de la discrétisation.

Citons à titre d'exemple le schéma de Milshtein pour le système (1) écrit au sens d'Ito, l'hypothèse de commutativité étant satisfaite :

$$\begin{aligned} \bar{X}_{p+1}^h &= \bar{X}_p^h + \sum_{j=1}^r s_j(\bar{X}_p^h) \Delta_{p+1}^h W^j + b(\bar{X}_p^h) h \\ &\quad + \sum_{j < k} \partial s_j(\bar{X}_p^h) s_k(\bar{X}_p^h) \Delta_{p+1}^h W^j \Delta_{p+1}^h W^k \\ &\quad + \frac{1}{2} \sum_{j=1}^r \partial s_j(\bar{X}_p^h) s_j(\bar{X}_p^h) \left[ (\Delta_{p+1}^h W^j)^2 - h \right] \end{aligned}$$

où on a défini les variables aléatoires gaussiennes  $\Delta_{p+1}^h W^j$  par :

$$\Delta_{p+1}^h W^j = W^j((p+1)h) - W^j(ph)$$

## 6 Mode d'emploi

L'utilisation de Presto ne nécessite aucun apprentissage, quelques contraintes doivent seules être prises en compte.

Les principales concernent les fenêtres de saisie algébrique : l'écriture d'une expression algébrique même relativement simple peut requérir plusieurs lignes de la fenêtre de saisie, par conséquent, pour le lecteur de Presto (identique, en ces occasions, à celui de Reduce), le caractère Return n'est pas le signal de fin de lecture, ce dernier étant le caractère “.”.

D'autre part, lors d'une telle saisie, il ne faut pas écrire “.1” mais “0.1”.

Ces contraintes sont rappelées à l'utilisateur chaque fois qu'elles sont nécessaires, dans le bandeau au-dessus de la fenêtre de saisie.

Signalons que les décimaux seront automatiquement transformés sous forme de fractions rationnelles (ainsi 0.1 sera transformé de manière interne par  $\frac{1}{10}$  : ceci est dû au fait que l'option “float” de Reduce n'est pas utilisée, car, dans le cas contraire, le lecteur transforme automatiquement toute fraction rationnelle en une valeur flottante approchée (ainsi  $\frac{1}{3}$  devient 0.333333, ce qui nuit à l'efficacité du simplificateur, et peut avoir des effets fâcheux sur l'exactitude des expressions algébriques calculées).

Les données saisies dans de telles fenêtres peuvent dépendre de paramètres non numériques (leur valeur, nécessaire à l'exécution du programme Fortran, sera demandée plus tard, et écrite dans le fichier de données Fortran lu par le programme lors de l'exécution ; il est donc possible de modifier ces valeurs sans avoir à engendrer un nouveau programme Fortran, ni même effectuer une nouvelle compilation).

Il existe une contrainte sur les noms de ces paramètres : ils ne doivent pas commencer par x, ni par t, ni par n, ni par dw ; ils ne peuvent pas être non plus pi ou h ou i.

Il convient enfin de respecter les conventions imposées par Fortran : un paramètre auquel on donnera des valeurs réelles ne devra pas porter un nom débutant par i, j, k, l, m.

En ce qui concerne les menus, certains présentent l'option “Quitter”. Si la question concerne une donnée (par exemple, le menu “Schéma Numérique” permet de choisir un schéma de discrétisation parmi une liste de schémas possibles), le clic sur “Quitter” aura l'effet suivant (sauf mention contraire ci-après, et outre la poursuite de la session) : si un choix avait été déjà effectué au cours de la session courante de Presto, ce choix est conservé ; sinon, la donnée sera considérée comme manquante, et signalée comme telle ultérieurement (il conviendra alors de la rentrer).

## 6.1 Le Menu Principal

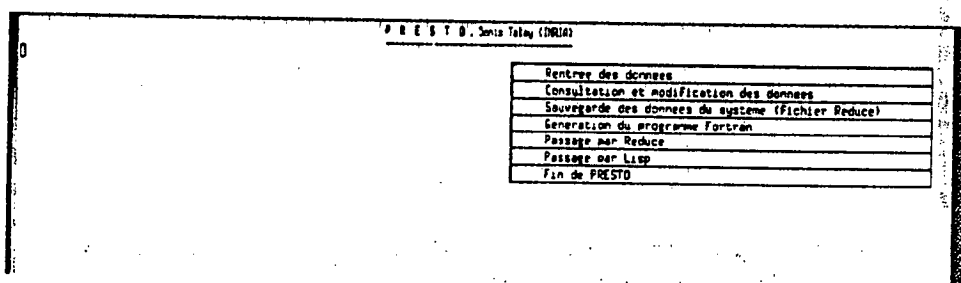
C'est le premier menu créé à l'ouverture d'une session Presto .

Ses 4 premières options permettent d'enchaîner sur une succession d'autres menus. A la fin de cet enchaînement on est systématiquement ramené au Menu Principal.

On peut aussi ouvrir des sessions Lisp et Reduce.

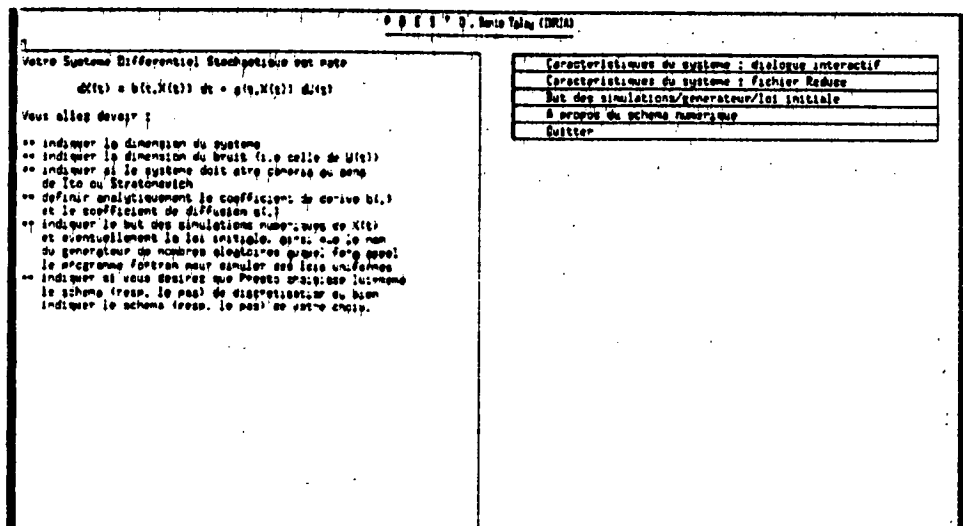
La dernière option permet de clôturer la session de Presto .

Tout ceci sera détaillé dans les paragraphes suivants.



En début de session, en général on clique sur l'option "Rentrée des données".

## 6.2 Rentrée des données



Les quatre premières options permettent de décrire totalement le problème de l'utilisateur, l'enchaînement des actions que provoque la sélection de chacune d'entre elles ramène toujours à ce menu, qui permet également de revenir au Menu Principal grâce à l'option "Quitter".

Pour décrire son problème, l'utilisateur doit sélectionner successivement : la première ou la deuxième option, puis la troisième et la quatrième.

Ainsi que nous le verrons plus loin, Presto signalera en temps utile toute description incomplète, il sera alors possible d'y remédier.

### 6.3 Caractéristiques du système : dialogue interactif

Si l'on a choisi de rentrer les données interactivement, on commence par écrire la dimension du système (1).

#### 6.3.1 Dimension du système

The screenshot shows a window titled "P R E S T O - Serie 1000 (IM21A)". Inside, there is a text box on the left containing the text "Saisie de la dimension de X12." and "La valeur actuelle ou par défaut est :". To the right of this text box is a control panel with three buttons: "Quitter sans modifier", "Modifier", and "Donner la valeur : 1".

Il est possible à tout moment de modifier les données déjà rentrées. Ceci permet, par exemple, d'engendrer successivement plusieurs programmes Fortran au cours d'une même session **Presto**.

Ceci explique le rappel de la valeur actuelle de la dimension. On retrouvera ce type de rappel en prélude à toute opération de saisie par clavier.

Bien entendu, "Donner la valeur ; 1" a le même effet que "Modifier" puis la rentrée de la valeur 1.

Examinons les conséquences d'un clic sur "Modifier".

Si l'on clique sur "Modifier", quatre fenêtres apparaissent. Les seuls événements admis par Presto sont le clic sur "Annuler" ou la frappe au clavier.

Le clic sur "Annuler" permet de conserver l'expression existante (cette possibilité était déjà offerte par le menu précédent, mais deux précautions valent mieux qu'une).

Une réponse non entière ou nulle provoque l'apparition d'une cinquième fenêtre, qui indique l'erreur ; le clic sur "Valider" qui est attendu (tout autre événement n'a aucun effet) va provoquer la réapparition de la fenêtre de saisie, nettoyée de la réponse rejetée.

The screenshot shows the Presto software interface with two windows. The left window is titled "Saisie de la dimension de X(t)." and contains a large text area with a small "ANNULER" button at the top right. The right window is titled "La valeur actuelle ou par défaut est :" and contains a large text area. Below the left window, there is a "VALIDER" button and a message box that reads: "Il faut un entier strictement positif. La valeur actuelle est conservée. Cliquez sur Valider."



### 6.3.2 Dimension du bruit

De même, on rentrera la dimension du bruit, i.e celle de  $(W(t))$ .

P R O G R A M M E - Dimension du bruit (DPR1A)

Saisie de la dimension de  $W(t)$ .  
La valeur actuelle ou par défaut est :

Quitter sans modifier  
Modifier  
Donner la valeur : 1

A ce stade, on définit algébriquement les coefficients de (1).

### 6.3.3 Rentrée des coefficients

P R O G R A M M E - Rentrée des coefficients (DPR1B)

À présent, vous allez devoir définir algébriquement les coefficients de dérive et de diffusion du système. Veuillez cliquer.

Fonction  $b(t,...)$   
Matrice de diffusion  $a(t,...)$   
Quitter

Le clic sur "Quitter" initialise les coefficients à 0, ou des vecteurs nuls, ou des matrices nulles, selon les dimensions respectives de  $(X(t))$  et  $(W(t))$ .

Les fonctions coordonnées de *b* et *s* sont saisies une à une.

On a pu éventuellement leur donner une valeur, préalablement, au cours de la session courante de Presto. Cette valeur est rappelée, on peut éventuellement la modifier (cf la saisie des dimensions).

Si on a choisi de les modifier, le menu suivant apparaît :

The screenshot shows a window titled "P R E S T O - Denis Volay (DMC)". It contains two main panels. The left panel has a title bar "Saisie de la coordonnée 2 du vecteur bit.4.2)." and contains the following text: "Votre réponse peut occuper plusieurs lignes ; elle doit IMPÉRATIVEMENT se terminer par un POINT-VIRGULE. Ne pas taper .1 mais 0.1". Below this is a text area containing the code: `sin(x)=2)/(1.2*x)-2*x^2)  
-cos(alpha*x)*pi*x2)  
+0.13*gamma*log(1+x)  
-sqrt(2*sin(x)) ;]`. The right panel has a title bar "La valeur actuelle ou par défaut est :" and contains a large empty text area.

Comme il est expliqué au début de ce paragraphe, les fenêtres de saisie algébrique nécessitent quelques précautions.

Si l'on commet une erreur de syntaxe (par exemple, on oublie de fermer une parenthèse, ou bien on tape ".12" au lieu de "0.12"), les conséquences sont imprévisibles ; en général, on se retrouve au top-level de *Le\_Lisp*, et le mieux est d'essayer la procédure "relancement" décrite en fin de manuel.

Ceci encourage à créer soi-même, sous un éditeur de texte, ses propres fichiers de données sur le modèle de ceux créés par Presto (un exemple est donné plus loin), et à les charger par l'option que nous allons décrire à présent.

## 6.4 Caractéristiques du système : fichier Reduce

Les données concernant (1) peuvent avoir été préalablement écrites ou sauvegardées dans un fichier lisible par Reduce (voir plus loin pour le contenu d'un tel fichier).

Dans ce cas, il suffit d'avoir cliqué sur l'option "Caractéristiques du système : fichier Reduce" du menu "Rentrée des données" pour voir le menu suivant.

PRESTO. Menu Fichier (FDR10)

Saisie du nom du fichier Reduce à charger.  
La valeur actuelle ou par défaut est :  
data.red

Quitter sans modifier  
Modifier  
Donner la valeur : data.red

La première et la dernière option permettent un chargement immédiat du fichier correspondant. On n'a alors à rentrer ni les dimensions de  $(X(t))$  et  $(W(t))$ , ni les expressions analytiques des coefficients.

L'option "Modifier" permet de changer le nom du fichier à charger. Un nom relatif par rapport à la directory de laquelle a été appelé Presto est autorisé.

Si le fichier n'existe pas, un message d'erreur apparaît, et son nom est à nouveau demandé.

## 6.5 Ito ou Stratonovich

P R E S T O . Sous Taly (DIRIG)	
Votre Systeme Differentiel Stochastique est-il defini au sens de Ito ou de Stratonovich ? Veuillez cliquer.	Equation au sens d'Ito
	Equation au sens de Stratonovich
	Quitter

Ainsi qu'il est expliqué plus haut, en cas de clic sur l'option "Stratonovich", Presto réécrit le système (1) au sens d'Ito.

Le clic sur "Quitter" équivaut à la conservation de la réponse précédente (si on est déjà passé par ce menu au cours de la session courante), sinon Presto considérera la donnée manquante.

## 6.6 But des simulations

P R E S T O . Sous Taly (DIRIG)	
Saisie du but de la discretisation.	Simulation de trajectoires
	Calcul de $E(f(X(t)))$ en temps fini
	Calcul de $\langle f(x), m(x) \rangle$ , $m$ loi invariante de $(X(t))$
	Calcul d'exposant de Lyapounov
	Quitter

Cette étape est cruciale, tant pour la suite du dialogue que pour le choix de l'algorithme de discrétisation.

Presto ne fait pas de vérification théorique : on peut donc engendrer un programme "simulant" des trajectoires même si la condition de commutativité n'est pas satisfaite (dans ce cas, les algorithmes divergent), ou calculant  $\frac{1}{N} \sum_{p=1}^N f(\bar{X}_p^h)$  pour  $N$  de plus en plus grand ( $\bar{X}_p^h$  étant le processus approchant  $(X(t))$ ), alors que le processus  $(X(t))$  n'est pas ergodique.

## 6.7 Temps d'intégration numérique.

Cette quantité est demandée même pour les calculs de type ergodique, ce qui permet à l'utilisateur de contrôler le temps de calcul. Elle sera écrite dans le fichier de données Fortran. On pourra donc la modifier sans avoir à recompiler le programme Fortran.

Il est donc prudent de la fixer d'abord petite, quitte à exécuter le programme une seconde fois...

P R E S T O : Dans l'alg (1982)	
Saisie du temps d'intégration numérique. La valeur actuelle ou par défaut est :	Quitter sans modifier
	Modifier
	Donner la valeur : 1

## 6.8 Loi initiale

Ce menu n'est pas abordé pour les calculs de type ergodique (calcul de loi invariante ou d'exposant de Lyapunov), la loi initiale est alors fixée par Presto à une masse de Dirac en un point déterministe de la sphère-unité (pour des raisons algorithmiques).

P R E S T O : Dans l'alg (1982)	
Saisie de la loi de $X(0)$ .	Déterministe
	La somme des masses réduite
	La somme quelconque
	Quitter

Si  $X(0)$  a été choisi déterministe, Presto saisit sa valeur numérique coordonnée par coordonnée (le clic sur "Modifier" crée une fenêtre de type algébrique, attention donc).

P R E S T O . Base Today (DRI4)				
Saisie de la coordonnée 1 du vecteur $X(0)$ . Votre réponse peut occuper plusieurs lignes : elle doit IMPÉRATIVEMENT se terminer par un POINT-VIRGULE. Ne pas taper .1 mais 0.1 La valeur actuelle ou par défaut est :	<table border="1"> <tr><td>Quitter sans modifier</td></tr> <tr><td>Modifier</td></tr> <tr><td>Donner la valeur : 0</td></tr> </table>	Quitter sans modifier	Modifier	Donner la valeur : 0
Quitter sans modifier				
Modifier				
Donner la valeur : 0				
0				

Si la loi initiale a été choisie gaussienne centrée réduite, Presto initialise l'espérance et la matrice de covariance en conséquence.

Sinon, elles sont saisies terme à terme (dans des fenêtres de type algébrique).

P R E S T O . Base Today (DRI4)				
Saisie de la coordonnée 1 du vecteur espérance de $X(0)$ . Votre réponse peut occuper plusieurs lignes : elle doit IMPÉRATIVEMENT se terminer par un POINT-VIRGULE. Ne pas taper .1 mais 0.1 La valeur actuelle ou par défaut est :	<table border="1"> <tr><td>Quitter sans modifier</td></tr> <tr><td>Modifier</td></tr> <tr><td>Donner la valeur : 0</td></tr> </table>	Quitter sans modifier	Modifier	Donner la valeur : 0
Quitter sans modifier				
Modifier				
Donner la valeur : 0				
0				

Lors de sa saisie, il est tenu compte du fait que la matrice de covariance est symétrique. Par contre, aucune vérification n'est faite concernant la positivité de ses coefficients, ou l'ordre de grandeur des termes diagonaux par rapport aux autres. Des erreurs à ce niveau peuvent provoquer une interruption de la session lors de la génération du programme Fortran, il restera alors à espérer au succès de la procédure "relancement" décrite plus loin.

Saisie du terme (1..1) de la matrice coherance de XIC. votre reponse peut occurer plusieurs lignes : elle doit IMPERATIVEMENT se terminer par un POINT-VIRGULE. Ne pas taper .1 mais 0.1 La valeur actuelle ou sa* defaut est :		Quitter sans modifier Modifier Donner la valeur : 0
--	--	---

## 6.9 Générateur de nombres aléatoires

La programmation Fortran de la simulation de lois gaussiennes ou discrètes dépend essentiellement de la syntaxe d'appel au générateur de nombres aléatoires. Pour l'instant, sont pris en compte le générateur de la bibliothèque NAG, et celui de Fortran 77 sous Unix. Cette liste peut être adaptée sans problème aux besoins de l'utilisateur.

<p>SECRET - Service Policy (DPR)</p>	
<p>Service du type de generateur de 10: 4. Forme utilisee.</p>	<p>LIBRARY NO.</p> <p>LIBRARY Form 872</p> <p>Quitter</p>

## 6.10 Fonction à intégrer

Ce menu apparaît lorsque l'on veut intégrer une fonction  $f$  par rapport à la loi de  $X(t)$  pour tout  $t$  dans l'intervalle d'intégration choisi (i.e on veut calculer  $Ef(X(t))$ ), ou bien par rapport à la loi invariante de  $(X(t))$ .

La saisie s'effectue de la même manière que celle des coefficients  $b$  et  $s$ . Si on modifie la valeur courante de  $f$ , on écrit la nouvelle valeur dans une fenêtre de saisie algébrique.

F I S T O : Saisie 'aleg' (J2014)	
Saisie de la coordonnée 1 du vecteur $f(t, a_1, a_2)$ . Votre réponse peut occuper plusieurs lignes ; elle doit IMPÉRATIVEMENT se terminer par un POINT-VIRGULE. Ne pas taper .1 mais 0.1 La valeur actuelle ou par défaut est :	Quitter sans modifier
	Modifier
	Donner la valeur : 0



## 6.11 Nombre de simulations

The screenshot shows a window titled "P R E S T O - Serie Toly (FORTRAN)". On the left, a text box contains the prompt "Saisie de nombre de simulations." and is currently empty. On the right, there is a menu with three options: "Nombre de simulations choisi par vous", "Nombre de simulations par défaut", and "Quitter".

Ce menu apparaît lorsque l'on veut calculer  $Ef(X(t))$  : Presto met en oeuvre une méthode de Monte-Carlo, et l'utilisateur peut contrôler le nombre de simulations indépendantes du processus approchant  $(X(t))$ .

Ce nombre est écrit dans le fichier de données Fortran, mais il sert également au dimensionnement d'un tableau dans le programme (Fortran n'acceptant pas le dimensionnement dynamique des tableaux). Il est important de lui donner a priori une grande valeur (qui servira seulement, lors de l'exécution du programme Fortran, à réserver une place mémoire suffisante), quitte à réduire ensuite cette valeur par édition du fichier de données Fortran.

Actuellement, Presto n'offre pas de moyen de contrôler si le choix est raisonnable (par mise en oeuvre du théorème de limite centrale par exemple). Ceci devrait évoluer.

La saisie est semblable à celle de la dimension du système. La valeur par défaut est 1000.

This screenshot shows the same Presto window, but with the value "1000" entered into the text box on the left. The menu on the right now has three options: "Quitter sans modifier", "Modifier", and "Donner la valeur : 1000".

## 6.12 Schéma de discrétisation

La dernière option du menu "Rentrée des données" concerne la méthode de discrétisation de (1) qui sera mise en oeuvre numériquement.

The screenshot shows the Presto software interface with the title bar "P R E S T O - Sous Talay (DRI2)". The main window is titled "Saisie du schéma numérique." and contains a list of numerical schemes:

Schéma choisi par Presto
Schéma d'Euler
Schéma de Milstein
Schéma Monte-Carlo
Schéma MCSM
Quitter

La base de connaissance de Presto lui permet de sélectionner une méthode de discrétisation dont la vitesse de convergence pour le but indiqué est élevé.

Toutefois, l'utilisateur a la liberté d'imposer son propre choix. D'une part, ceci est indispensable à des fins de test ou de mise au point, d'autre part on trouvera dans Talay [34] un exemple où le schéma d'Euler donne des résultats au moins aussi bons que des schémas réputés bien meilleurs par la théorie (c'est précisément cet exemple qui est programmé au paragraphe (4)).

## 6.13 Pas de discrétisation

Enfin, on a à fixer le pas de discrétisation. Cette valeur sera écrite dans le fichier de données Fortran. La valeur par défaut est 0.1.

The screenshot shows the Presto software interface with the title bar "P R E S T O - Sous Talay (DRI2)". The main window is titled "Saisie du pas de discrétisation." and contains a form for entering the discretization step:

Pas de discrétisation fixe par vous
Valeur par défaut
Quitter

## 6.14 Génération du programme Fortran

Lorsque toutes les données précédentes ont été rentrées, on revient au Menu Principal (en quittant le menu "Rentrée des données").

Après avoir cliqué sur "Génération du programme Fortran", on trouve un premier menu qui permet de préciser si les constantes du programme Fortran devront être engendrées en simple ou double précision.

The screenshot shows a window titled "PRESTO - Service 'Ajout' (COM12)". Inside, there is a text box on the left with the message: "Les constantes devront-elles être écrites en simple ou double précision ? Veuillez cliquer." To the right of this text box is a button menu with three options: "Simple Precision", "Double Precision", and "Quitter".

Les données manquant à Presto pour la génération (parce qu'on a oublié de les rentrer) sont énumérées.

Le clic sur "Valider" ramène au Menu Principal, qui permet de compléter les données (voir ci-après).

The screenshot shows a window titled "PRESTO - Service 'Ajout' (COM12)". Inside, there is a text box on the left with the message: "Génération impossible : les données suivantes sont manquantes : pas de discrétisation. Cliquez sur Valider." To the right of this text box is a button labeled "VALIDER".

## 6.15 Consultation et modification

L'option "Consultation et modification des données" du Menu Principal crée une fenêtre dans laquelle sont rappelées les principales données, et un menu de sélection des données.

La remarque de la fenêtre inférieure est importante.

L'option "Quitter" ramène au Menu Principal.

P R E S T O - Denis Volny (1982/83)								
<p>Votre Systeme Differentiel Stochastique est note</p> $dX(t) = b(t, X(t)) dt + \sigma(t, X(t)) du(t)$ <p>Il est defini au sens de : Ito</p> <p>La dimension de X(t) est : 2</p> <p>La dimension de U(t) est : 1</p> <p>Le but des simulations est : calcul-generance-en-temps-fin</p> <p>La condition initiale est : gaussienne</p> <p>Le schema retenu est : schema-euler-maruyama</p>	<table border="1"><tr><td>Voir/modifier la dimension de X(t)</td></tr><tr><td>Voir/modifier la dimension de U(t)</td></tr><tr><td>Voir/modifier : simulations/generateur/loi X(0)</td></tr><tr><td>Voir/modifier : schema/pas d'integration</td></tr><tr><td>Voir/modifier le coefficient de derive b</td></tr><tr><td>Voir/modifier le coefficient de diffusion sigma</td></tr><tr><td>Quitter</td></tr></table>	Voir/modifier la dimension de X(t)	Voir/modifier la dimension de U(t)	Voir/modifier : simulations/generateur/loi X(0)	Voir/modifier : schema/pas d'integration	Voir/modifier le coefficient de derive b	Voir/modifier le coefficient de diffusion sigma	Quitter
Voir/modifier la dimension de X(t)								
Voir/modifier la dimension de U(t)								
Voir/modifier : simulations/generateur/loi X(0)								
Voir/modifier : schema/pas d'integration								
Voir/modifier le coefficient de derive b								
Voir/modifier le coefficient de diffusion sigma								
Quitter								
<p>Remarque : les changements de dimension provoquent AUTOMATIQUEMENT des changements des expressions algebriques des coefficients b(.) et sigma(.) (considerees respectivement comme vecteur et matrice) La reduction d'une dimension provoque la suppression de leurs dernieres lignes ou colonnes. L'augmentation d'une dimension cree des termes nuls supplementaires.</p>								

## 6.16 Paramètres

Eventuellement, on voit ensuite apparaître la fenêtre suivante.

En effet, Presto dispose d'un analyseur de formules qui lui permet de construire la liste de tous les paramètres que l'utilisateur a utilisés lors des saisies de type algébrique. (Cette opération n'est pas aussi simple qu'elle en a l'air, même si elle n'est pas très compliquée : le code interne de Reduce comprend une fonction qui permet d'isoler les noyaux d'une expression algébrique ; la liste rendue par cette fonction appliquée à

$$x * \sin(z) * \sin(3 * y + 2)$$

est

$$(x, \sin(z), y)$$

ce qui n'est pas satisfaisant (dans le fichier de données Fortran, c'est la valeur de  $z$  qui doit figurer, non le sinus de cette valeur) ; d'autre part, construire la liste de tous les symboles conduirait à considérer "sin" comme un paramètre...

La valeur de chacun des paramètres sera demandée (la fenêtre de saisie n'est pas de type algébrique, aucune précaution particulière n'est donc à prendre).

## 6.17 Fichiers Fortran

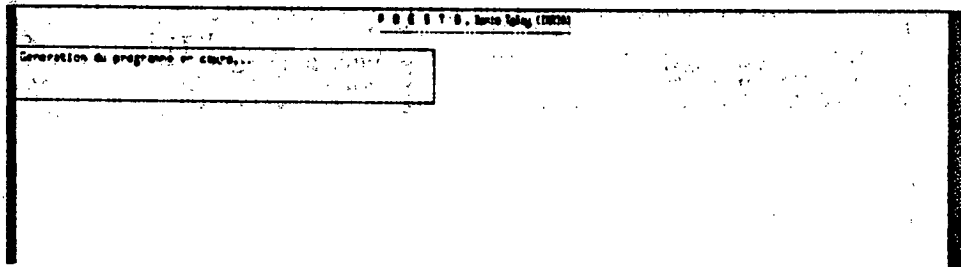
Presto demande le nom du fichier dans lequel sera écrit le programme Fortran.

P R E S T O - Serie Telsa (PDBIA)	
Saisie du nom du fichier qui contiendra le programme Fortran. La valeur actuelle ou par défaut est :	Quitter sans modifier Modifier Donner la valeur : org.f
org.f	

Presto crée le fichier de données Fortran, puis, lors de l'exécution, le programme Fortran créera un fichier de résultats. Les noms de ces fichiers sont à spécifier.

P R E S T O - Serie Telsa (PDBIA)	
Saisie du nom du fichier qui contiendra les paramètres du programme Fortran. La valeur actuelle ou par défaut est :	Quitter sans modifier Modifier Donner la valeur : org.data
org.data	

Si tout va bien, le programme peut être engendré. Après quoi, on sera ramené automatiquement au Menu Principal, à partir duquel on pourra rentrer de nouvelles données et engendrer de nouveaux programmes, ou quitter la session courante définitivement.



## 6.18 Sauvegarde des données

Une option du Menu Principal permet de sauvegarder la description du système (1) dans un fichier de sauvegarde, dont le nom est demandé.

P.O.T.T.O. Menu (FONCTION)	
<p>Don du fichier Reduct de sauvegarde. La valeur actuelle ou par défaut est :</p> <p>data.red</p>	<p>Quitter sans modifier</p> <p>Modifier</p> <p>Donner la valeur : data.red</p>

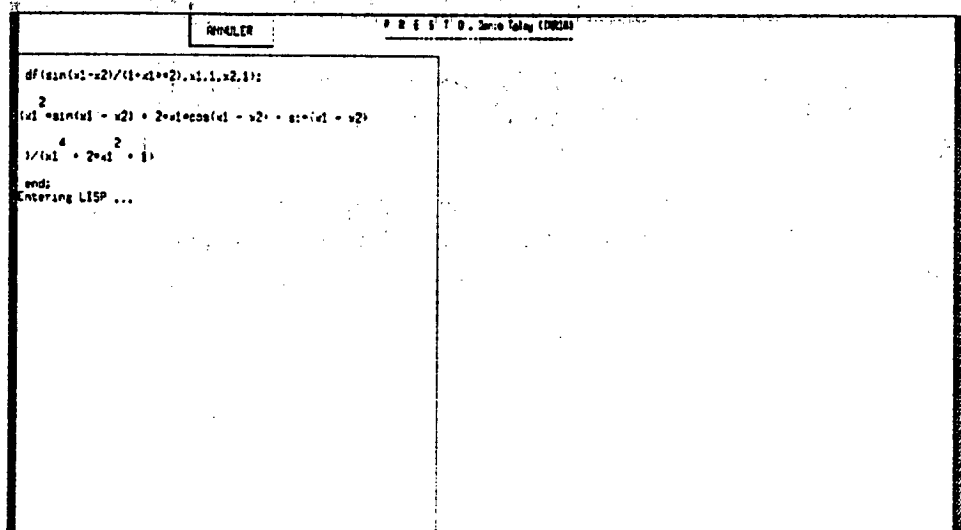
Par exemple, la sauvegarde des données correspondant au programme Fortran du paragraphe (4) produit le fichier suivant :

```
off echo$
off msg$
!$dimx:=2$
!$dimw:=2$
matrix !$b(!$dimx,1),!$sigma(!$dimx,!$dimw)$
!$b(1,1):=(3*sqrt(2)*x1 - 2*x1*sin(omega*t)*omega
- 12*x1 + 6*sqrt(2)*x2 - 6*x2)/(4*(cos(omega*t) + 2))$
!$b(2,1):=(6*sqrt(2)*x1 - 6*x1 + 3*sqrt(2)*x2
- 2*x2*sin(omega*t)*omega - 12*x2)/(4*(cos(omega*t) + 2))$
!$sigma(1,1):=sin(vnu*(x1 + x2))$
!$sigma(1,2):=cos(vnu*(x1 + x2))$
!$sigma(2,1):=sin((3*x1*vnu + 3*x2*vnu + pi)/3)$
!$sigma(2,2):=cos((3*x1*vnu + 3*x2*vnu + pi)/3)$
send$
```



## 6.19 Passage par Reduce et Lisp

Une option du Menu Principal permet d'ouvrir une fenêtre Reduce. On peut y effectuer des calculs algébriques, exactement comme lors d'une session ordinaire.



Il est fortement déconseillé de donner aux variables que l'on construit alors les noms "réservés" par Presto (c'est-à-dire débutant par x,t,n,dw, ainsi que h), ou ceux des paramètres des expressions algébriques (la substitution du symbole par la valeur est effectuée systématiquement par l'évaluateur de Reduce).

Pour quitter cette fenêtre, il ne suffit pas (c'est la seule exception) de cliquer sur "Annuler" : il faut d'abord repasser sous l'interprète de Lisp par la commande Reduce "end;", le clic sur "Valider" permet alors de revenir au Menu principal.

De la même manière, il est possible de créer une fenêtre Lisp, que l'on quitte cette fois par un simple clic sur "Valider".

## 6.20 Procédure de relancement

En cas d'erreur grave (d'un des types signalés précédemment), on se retrouve au top-level de *Le\_Lisp*. Le prompt ">" est en principe visible (sous X, il peut arriver qu'il soit masqué par une fenêtre).

Il faut alors essayer de taper "(top)" (sans les guillemets), ce qui permet le retour au menu principal, les données préalablement rentrées ne sont pas perdues, mais rien ne garantit d'effets de bord pervers.

Si, après avoir tapé "(top)", le menu principal n'apparaît pas rapidement, il n'y a pas d'autre solution que de supprimer la fenêtre à laquelle est attachée la session courante de Presto ... et de recommencer une nouvelle session.

## Références

- [1] T.A.AVERINA & S.ARTEMIEV : publications of the Siberian Academy of Sciences, 1987, 1988.
- [2] L.ARNOLD : *Stochastic Differential Equations*, J.Wiley, 1974.
- [3] L.ARNOLD, E.OELJEKLAUS & E.PARDOUX : Almost sure and moment stability for linear Ito Equations , *Lyapunov Exponents* , L.Arnold & V.Wihstutz Ed. , Lecture Notes in Mathematics 1186, Springer, 1986.
- [4] P.BAXENDALE : Moment stability and large deviations for linear stochastic differential equations , à paraître.
- [5] R.N. BHATTACHARYA : On the functional central limit theorem and the law of the iterated logarithm for Markov processes, *Z. Wahrscheinlichkeitstheorie verw. Gebiete* 60 (1982), 185-201.
- [6] A.CALZOLARI & C.COSTANTINI & F.MARCHETTI : A confidence interval for Monte-Carlo methods with an application to simulation of obliquely reflecting brownian motion , (à paraître in *Stoch. Proc. and their Appl.*), 1988
- [7] P.CHANCELIER & C.GOMEZ & J.P.QUADRAT & A.SULEM : Un système-expert pour l'optimisation de systèmes dynamiques , Congrès INRIA d'Analyse Numérique , 1985.
- [8] J.M.C.CLARK & R.J.CAMERON : The maximum rate of convergence of discrete approximations for stochastic differential equations , *Stochastic Differential Systems - Filtering and Control*, B. Grigelionis (Ed.), Proceedings of the IFIP Working Conference, Vilnius, Lithuania, USSR, 1978, Lecture Notes in Control and Information Sciences 25, Springer-Verlag, 1980.
- [9] J.M.C.CLARK : An efficient approximation for a class of stochastic differential equations , *Advances in Filtering and Optimal Stochastic Control*, W. Fleming & L. Gorostiza (Eds.), Proceedings of the IFIP Working Conference, Cocoyoc, Mexico, 1982, Lecture Notes in Control and Information Sciences 42, Springer-Verlag, 1982.
- [10] H.DOSS : Liens entre équations différentielles stochastiques et ordinaires , *Ann. Inst. H.Poincaré*, 13, 1977.
- [11] P.FLORCHINGER & F.LE GLAND : (en preparation)
- [12] A.GERARDI & F.MARCHETTI & A.M.ROSA : Simulation of diffusions with boundary conditions , *Systems & Control Letters* 4, 253,1984.
- [13] N.IKEDA & S.WATANABE : *Stochastic Differential Equations and Diffusion Processes* , North-Holland, 1981.
- [14] H.J. KUSHNER : *Approximation and weak convergence methods for random processes* , M.I.T. Press, 1984.

- [15] J.LEBLOND & D.TALAY : Simulation of diffusion processes with Presto, building systems like Presto with Adagio , Proceedings of the 7th French-Belgian Statisticians Meeting, Rouen, France, 1986.
- [16] E.J.McSHANE : *Stochastic calculus and stochastic models*, , Academic Press, 1974.
- [17] G.N. MILSHTTEIN : Approximate integration of stochastic differential equations , *Theory of Probability and Applications*, 19, 557-562, 1974.
- [18] G.N.MILSHTEIN : A method of second-order accuracy integration of stochastic differential equations , *Theory of Probability and Applications*, 23, 1976.
- [19] G.N.MILSHTEIN : Weak approximation of solutions of systems of stochastic differential equations , *Theory of Probability and Applications*, 30, 750-766, 1985.
- [20] N.J.NEWTON : An asymptotically efficient difference formula for solving stochastic differential equations , *Stochastics* 19, 175-206, 1986.
- [21] E. PARDOUX : Stabilité du mouvement des pales d'hélicoptères dans le cas d'un écoulement de l'air turbulent , *Actes du Colloque "L'Automatique pour l'Aéronautique"* , Paris (1986)
- [22] E.PARDOUX & M.PIGNOL : Etude de la stabilité de la solution d'une E.D.S. bilinéaire à coefficients périodiques. Application au mouvement d'une pale d'hélicoptère , *Analysis and Optimization of Systems, Part 2* , A.Bensoussan & J.L.Lions (Eds.), *Lecture Notes in Control and Info. Sc.* 63, 92-103, 1984.
- [23] E.PARDOUX & D.TALAY : Discretization and simulation of S.D.E. , *Acta Applicandae Mathematicae* 3,23, 1985.
- [24] E.PARDOUX & D.TALAY : Stability of linear differential systems with parametric excitation , *Nonlinear Stochastic Dynamic Engineering Systems* , G.I. Schueller & F. Ziegler (Eds.), *Proceedings of the IUTAM Symposium*, Innsbruck, 1987, Springer-Verlag, 1988.
- [25] M.PIGNOL : *Stabilité stochastique des pales d'hélicoptère* , Thèse de 3ème Cycle, Université de Provence
- [26] E.PLATEN : An approximation method for a class of Ito processes, *Lietuvos Matemematikos Rinkiniys*, 1981.
- [27] E.PLATEN & W.WAGNER : On a Taylor formula for a class of Ito processes , *Probability and Mathematical Statistics* ,3, 37-51, 1982.
- [28] H.J.SUSMANN : On the gap between deterministic and stochastic ordinary differential equations , *Ann.Prob.* 6, 1978.
- [29] D.TALAY : Résolution trajectorielle et analyse numérique des équations différentielles stochastiques , *Stochastics*, 9, 275-306, 1983.

- [30] D.TALAY : Efficient numerical schemes for the approximation of expectations of functionals of S.D.E. , *Filtering and Control of Random Processes* , H.Korezlioglu & G.Mazziotto & J.Szipirglas (Eds.), Proceedings of the ENST-CNET Colloquium, Paris, 1983, Lecture Notes in Control and Information Sciences 61, Springer-Verlag, 1984.
- [31] D.TALAY : Discretisation d'une E.D.S. et calcul approché d'espérances de fonctionnelles de la solution , M2AN 20-1, 141, 1986.
- [32] D.TALAY : Second Order Discretization Schemes of Stochastic Differential Systems for the computation of the invariant law , INRIA Report 753, 1987 (et soumis pour publication)
- [33] D.TALAY : Approximation of upper Lyapunov exponents of bilinear stochastic differential systems, INRIA Report 965, 1989 (et soumis pour publication)
- [34] D.TALAY : Simulation and Numerical Analysis of Stochastic Differential Systems , *Effective Stochastic Analysis* , P.Krée & W.Wedig (Eds.) (à paraître)